

Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

2.º Ano/ 1.º Semestre

Unidade Curricular: Arquitetura de Dispositivos

Docente: David Jardim

FICHA DE TRABALHO 7

Exercícios:

1. Crie um projeto com o nome Ficha7.
2. Implemente um Menu de um jogo usando tipos enumerados
 - a. Adicione as seguintes opções:
 - i. START
 - ii. SAVEGAME
 - iii. LOADGAME
 - iv. HIGHSCORES
 - v. QUIT
 - b. Imprima todas as opções existentes no menu
 - c. Utilize a instrução de seleção IF – ELSE para escolher uma das opções do menu anterior e imprima uma mensagem adequada
 - d. Utilize a instrução de seleção SWITCH para escolher uma das opções do menu anterior e imprima uma mensagem adequada
3. Crie uma classe denominada por **Rational**:
 - a. Adicione dois atributos
 - i. numerator: int
 - ii. denominator: int
 - b. Implemente a sobreposição do método **equals**
 - c. Implemente a interface Comparable<Rational> para comparar dois números racionais
 - d. Implemente a sobreposição do método **hashCode**
 - e. Crie uma lista de números racionais e adicione várias instâncias de números racionais à lista
 - f. Ordene a lista segundo a ordem natural (utilizando a interface Comparable)
4. Crie uma classe denominada por **Student**:
 - a. Adicione dois atributos
 - i. number: int
 - ii. name: String
 - b. Implemente a sobreposição do método **equals**
 - c. Implemente a interface Comparable<Student> para comparar dois alunos
 - d. Implemente a sobreposição do método **hashCode**
 - e. Crie uma lista de alunos e adicione várias instâncias de alunos à lista

Cofinanciado por:

- f. Ordene a lista segundo a ordem natural (utilizando a interface *Comparable*)
 - g. Implemente uma classe ***StudentsComparatorByNumber*** para comparar alunos que implementa a interface *Comparator<Student>* e que compara os alunos tendo em conta o seu número de aluno
 - h. Implemente uma classe ***StudentsComparatorByName*** para comparar alunos que implementa a interface *Comparator<Student>* e que compara os alunos tendo em conta o seu nome
 - i. Ordene a lista segundo a ordem natural dos nomes (utilizando a interface *Comparator<Student>*)
5. Crie uma classe denominada por ***Course***:
- a. Adicione dois atributos
 - i. code: int
 - ii. name: string
 - iii. credits: int
 - iv. ano: int
 - b. Implemente a sobreposição do método ***equals***
 - c. Crie um mapa de disciplinas e adicione várias instâncias de disciplinas ao mapa
 - d. Verifique se o mapa contém uma determinada disciplina
 - e. Imprima todas as chaves (*keys*) existentes no mapa
 - f. Imprima todas as disciplinas (*values*) existentes no mapa
 - g. Imprima todos os conjuntos (*key, value*) existentes no mapa
 - h. Cada disciplina tem um atributo que é o ano, para cada ano existente crie uma lista que irá armazenar todas as disciplinas referentes a esse ano

Cofinanciado por:

